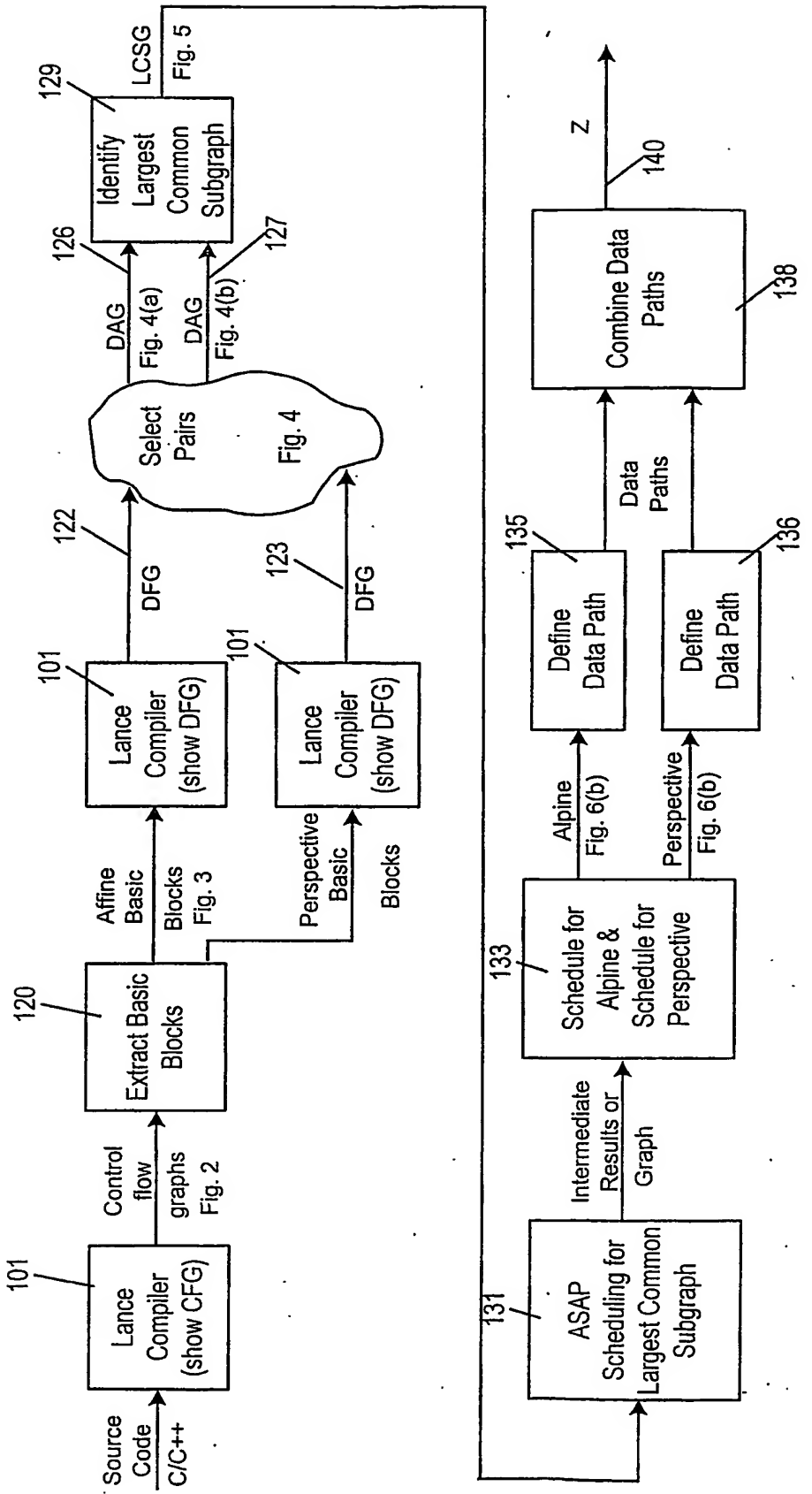


Fig. 1

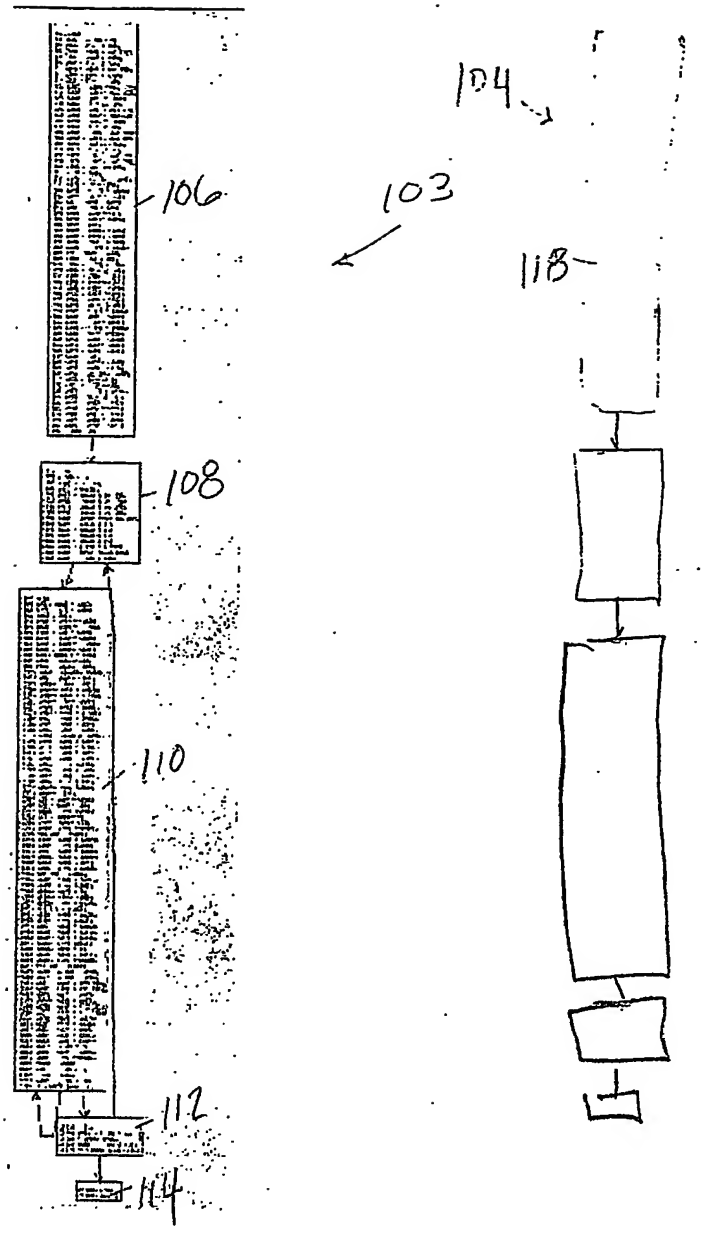


Fig. 2

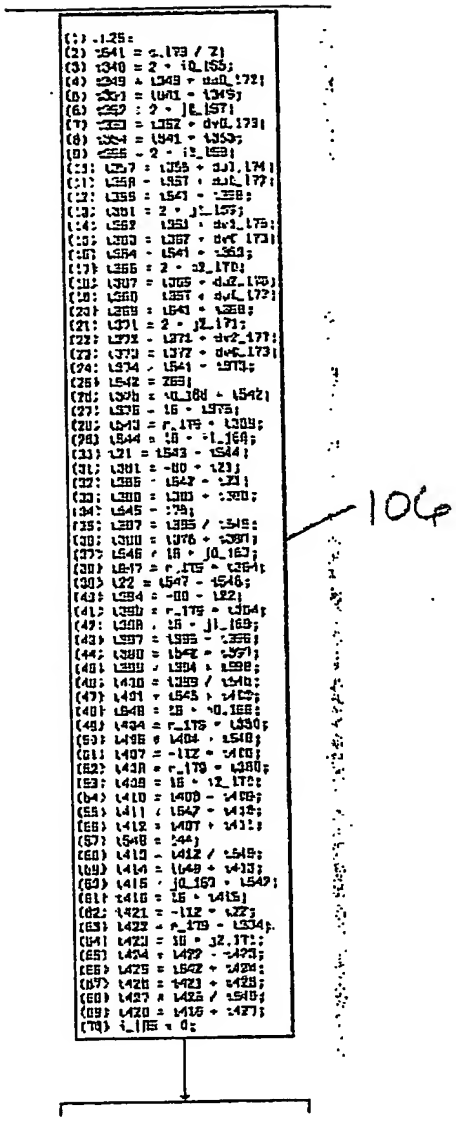


Fig. 3

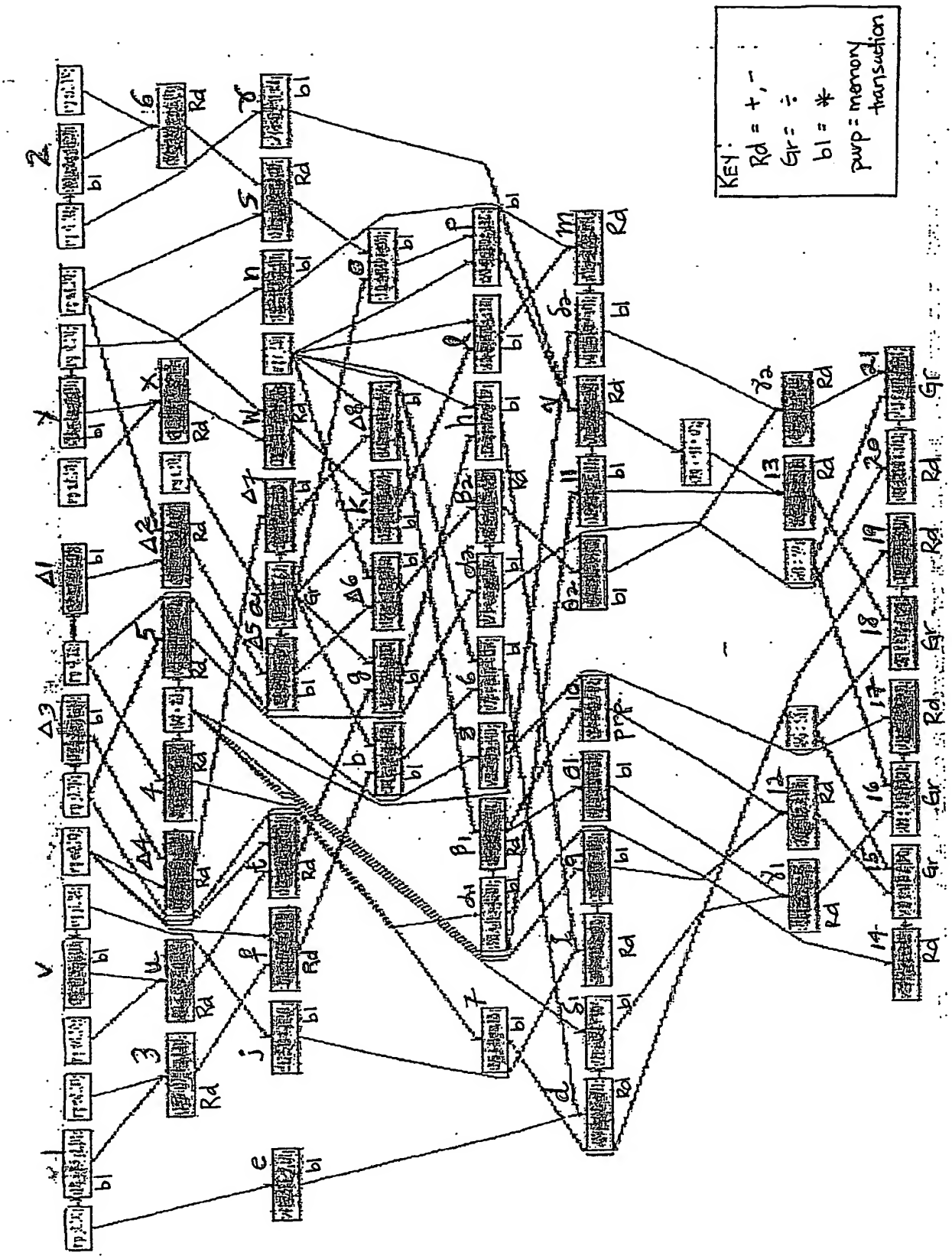


Fig. 4(a)

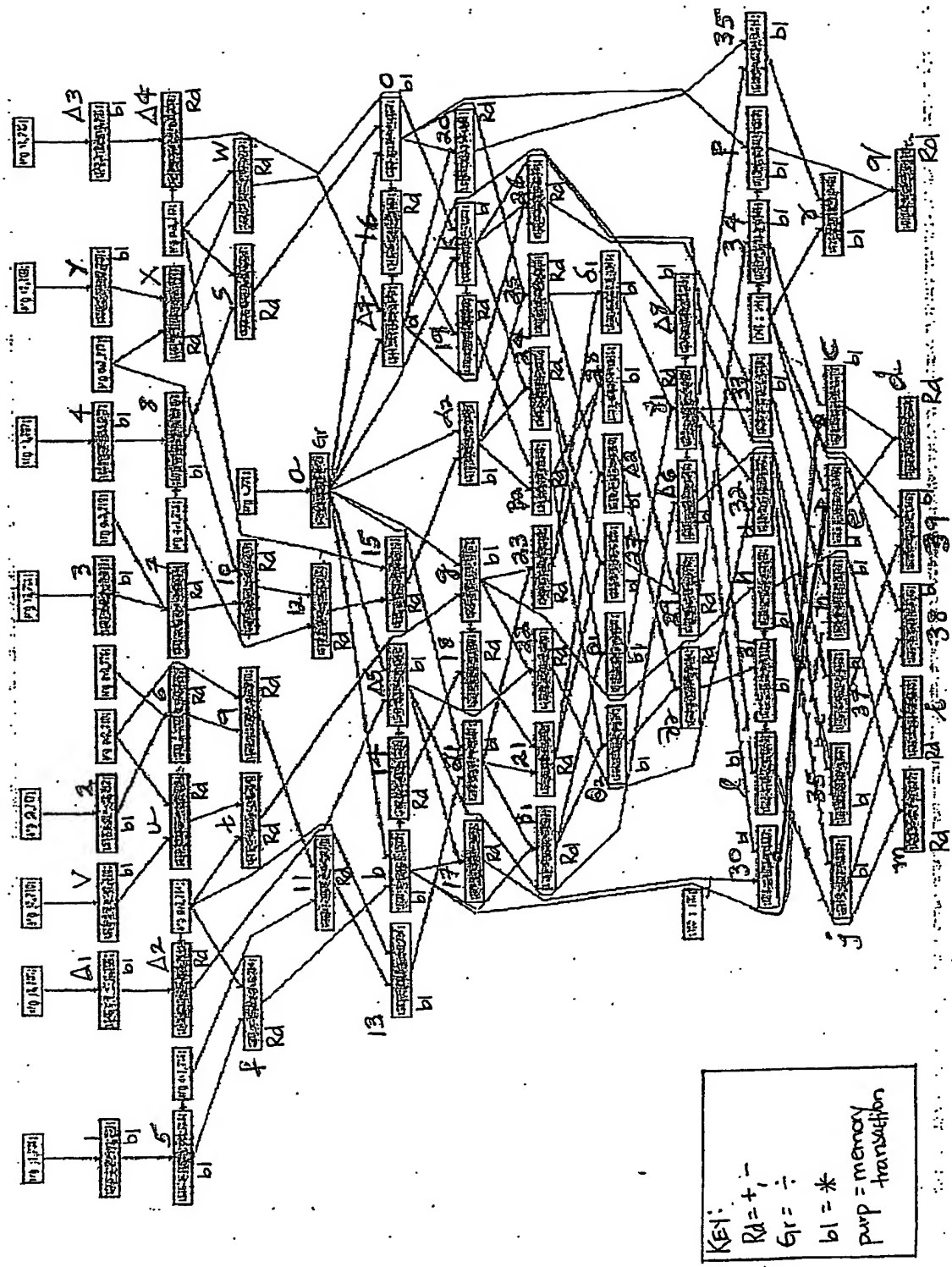


Fig. 4(b)

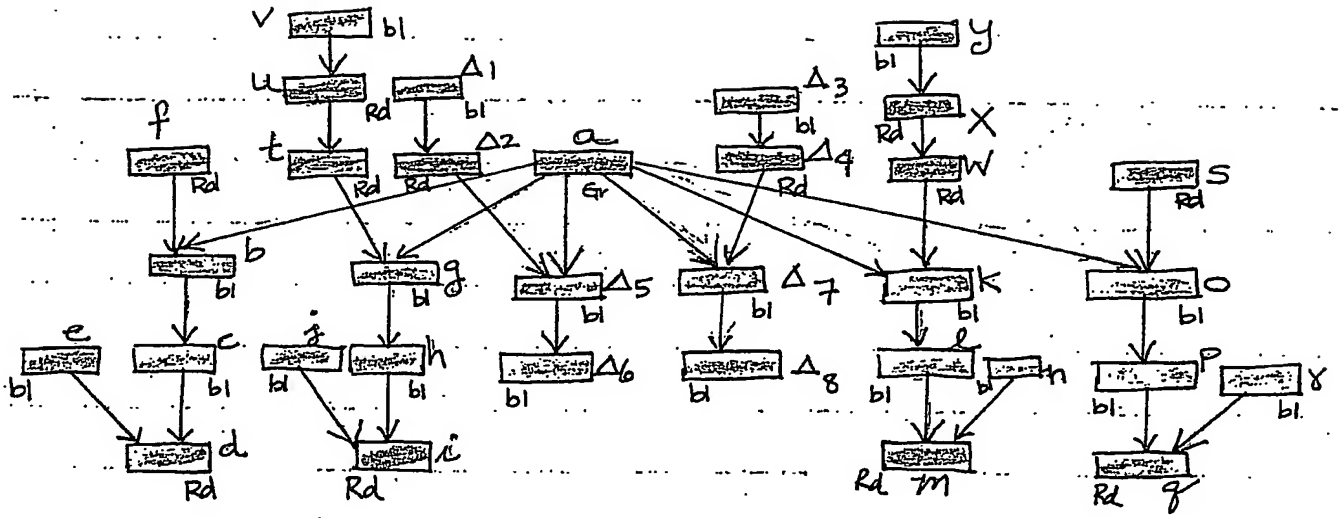


Fig. 5

KEY:
Rd = +, −
Gr = ÷
bl = *
purp = memory transaction

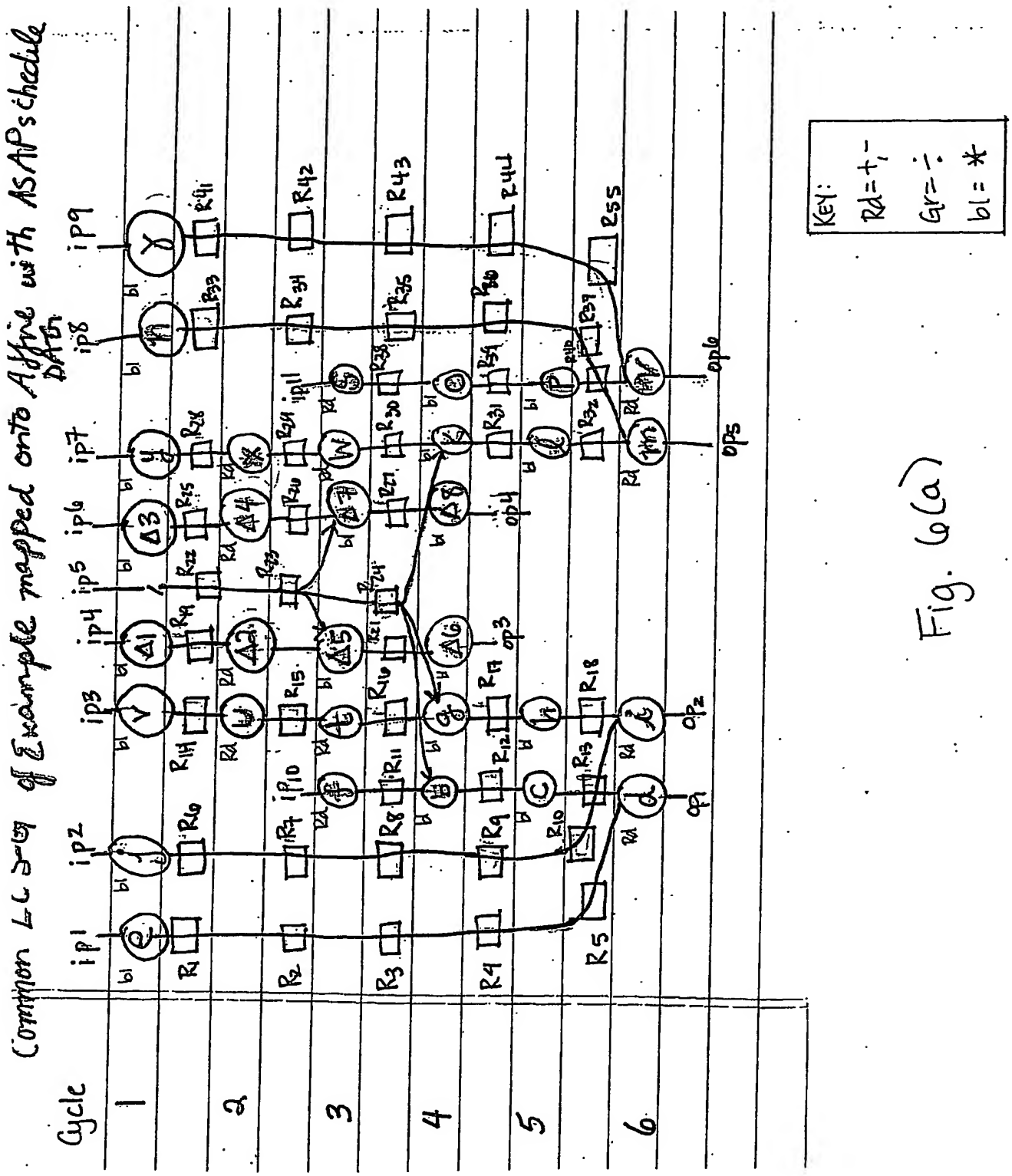

Common LC Seq of Example mapped onto Affine with ASAP schedule
DATA
Cycle
1
2
3
4
5
6
ip1
ip2
ip3
ip4
ip5
ip6
ip7
ip8
ip9
ip10
ip11
op1
op2
op3
op4
op5
op6
Key:
Rd=+,-
Gr=÷
bl=*

Fig. 6(a)

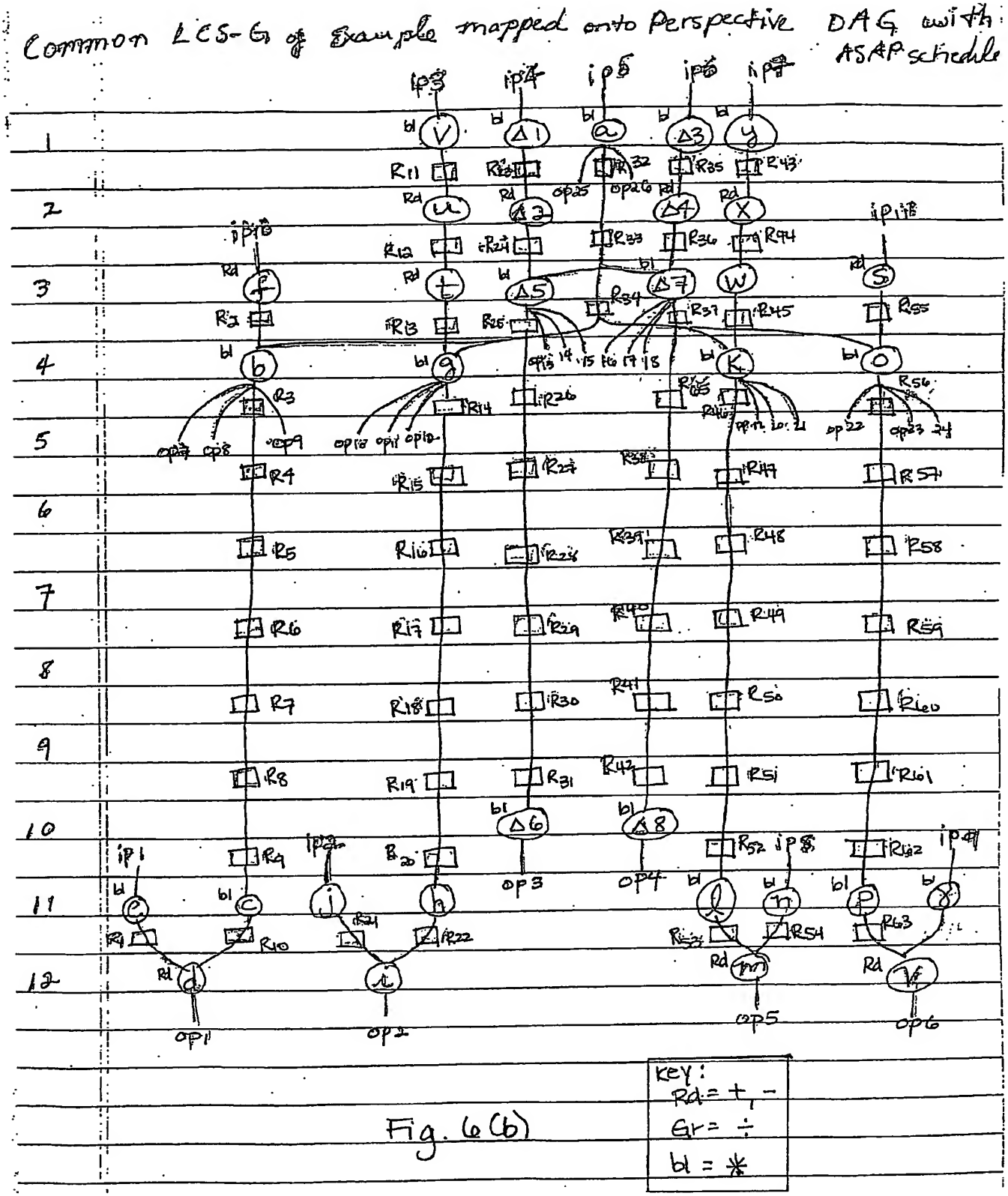

Common LCS-G of Example mapped onto Perspective DAG with ASAP schedule
key:
Rd = +, −
Gr = ÷
bl = *

Fig. 6(b)

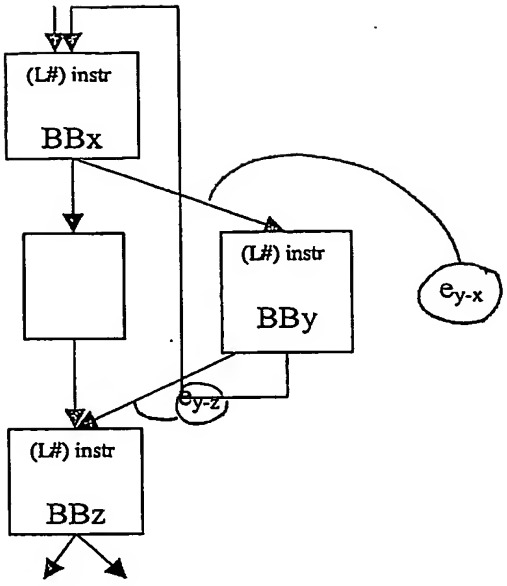


Fig. 7

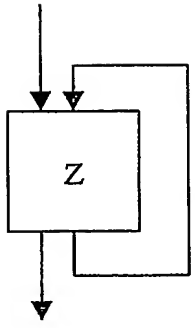


Fig. 8

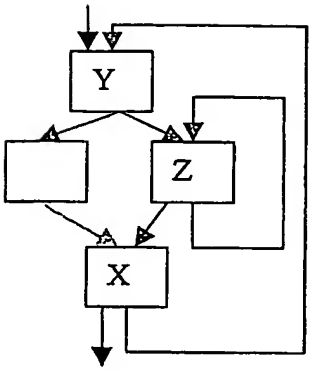


Fig. 9 (a)

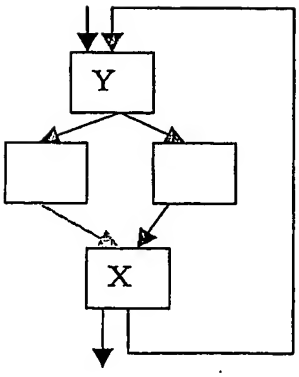


Fig. 9 (b)

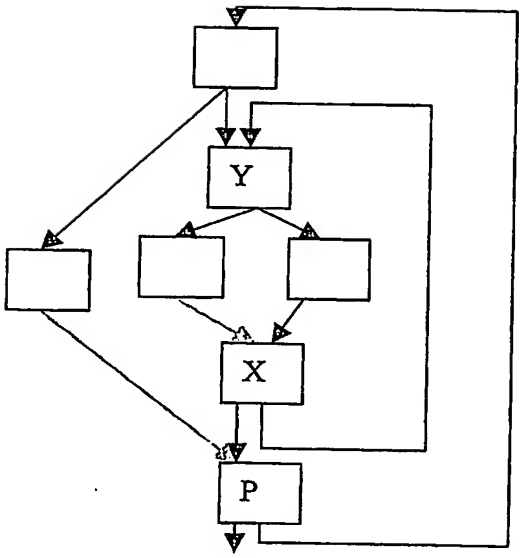
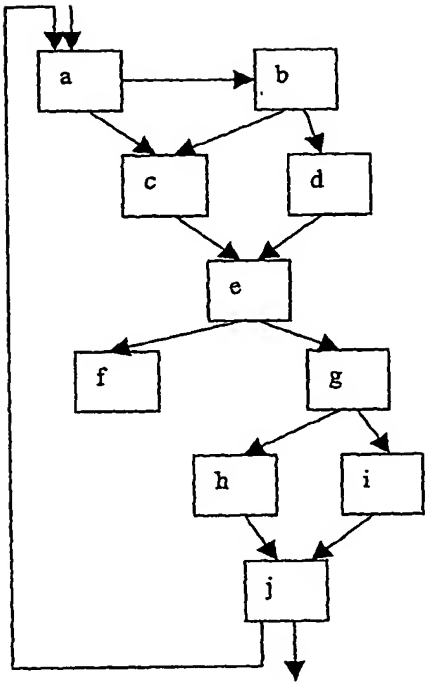


Fig. 10



Decision (a, b, e, g)
Merge (c, e, j)
Pass (d, f, h, i)

Fig. 11

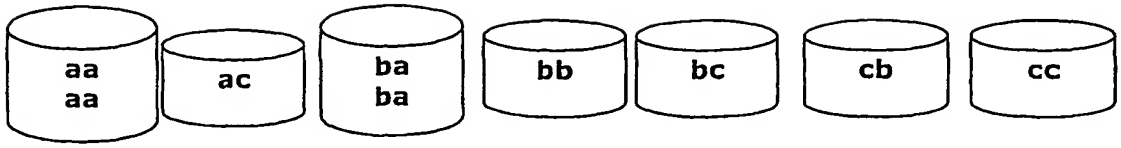
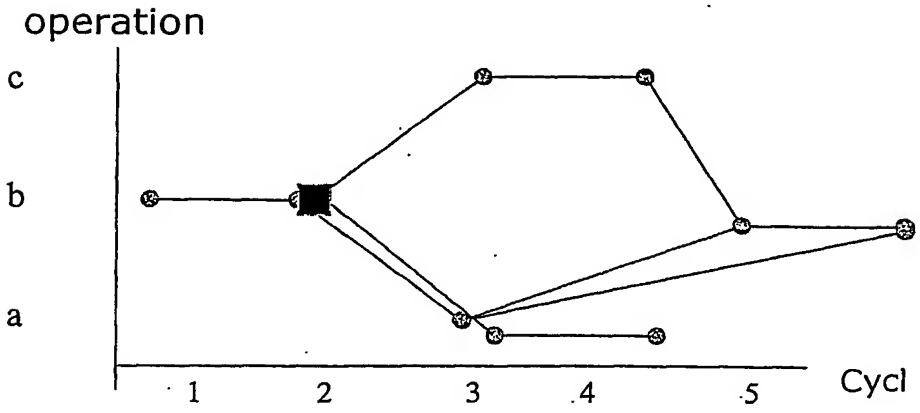
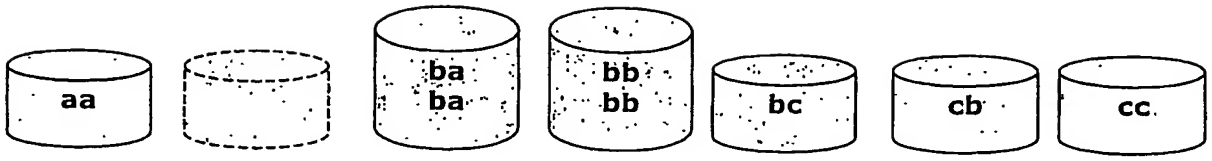


Figure 12: First Graph's edges arranged into a Bin Sequence



Graph number 2, Figure 13: The Second Graph



13(a)
Figure ~~14~~. The Second Graph arranged in a Modified Bin Sequence

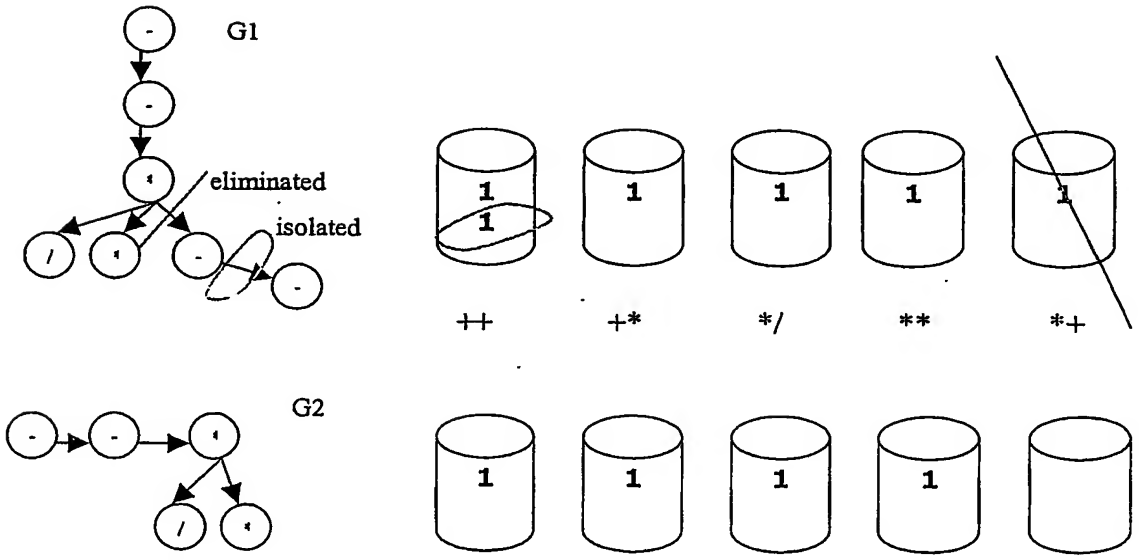


Fig. 14

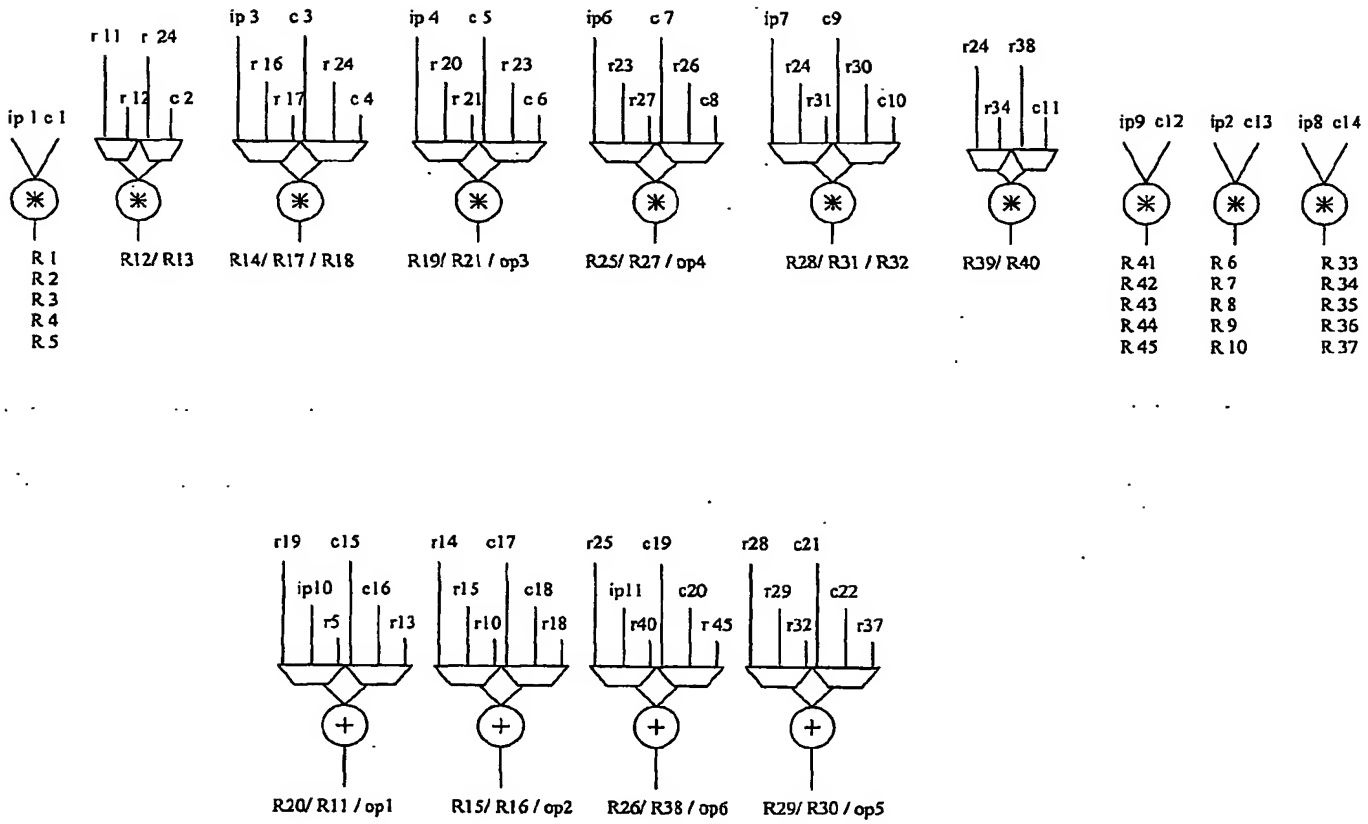
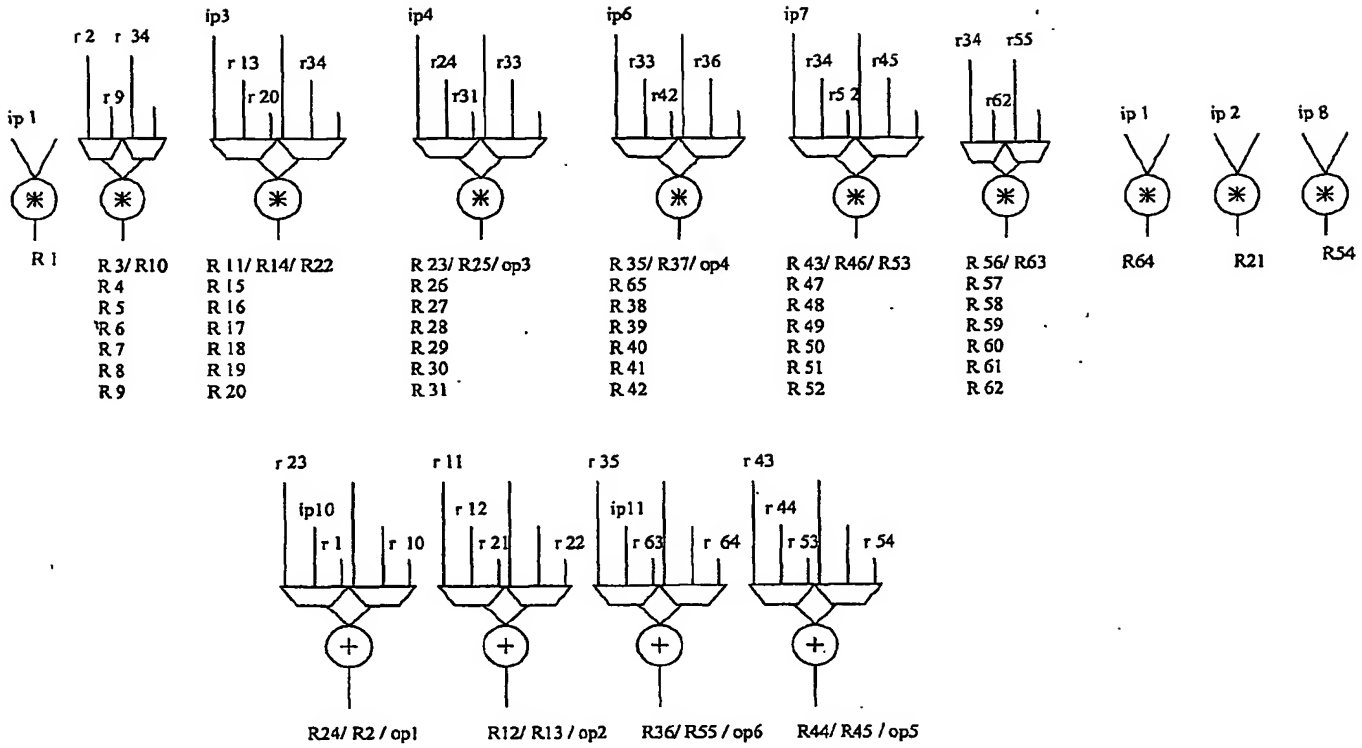
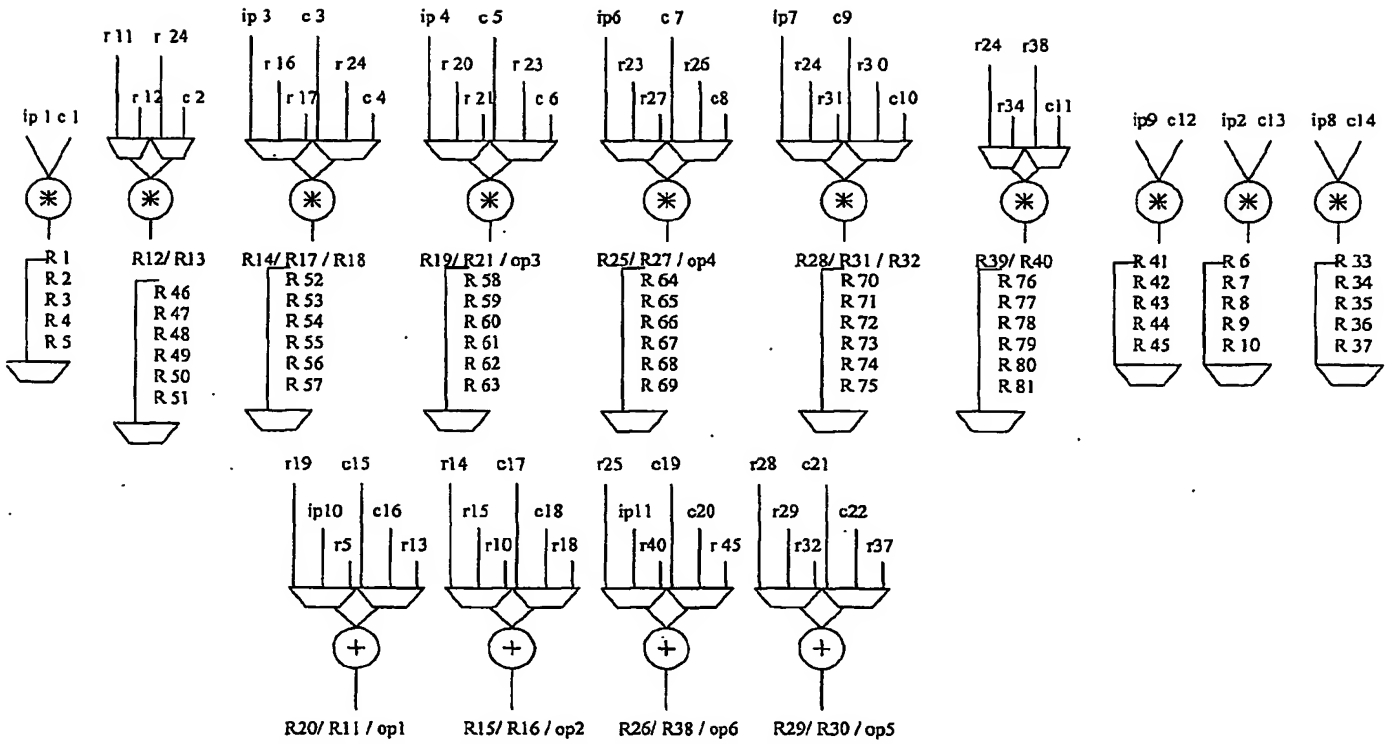


Fig:X Affine preloop common architecture after ASAP schedule



Note: All inputs unmarked indicate constants.

Figure 7 Perspective preloop common architecture after ASAP schedule



Note: Some multiplier and adder outputs also form outputs from the module.

Figure 2 Common architecture with multiplexers in delay paths to accommodate both affine and perspective instantiations.

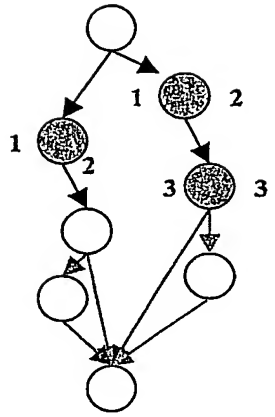
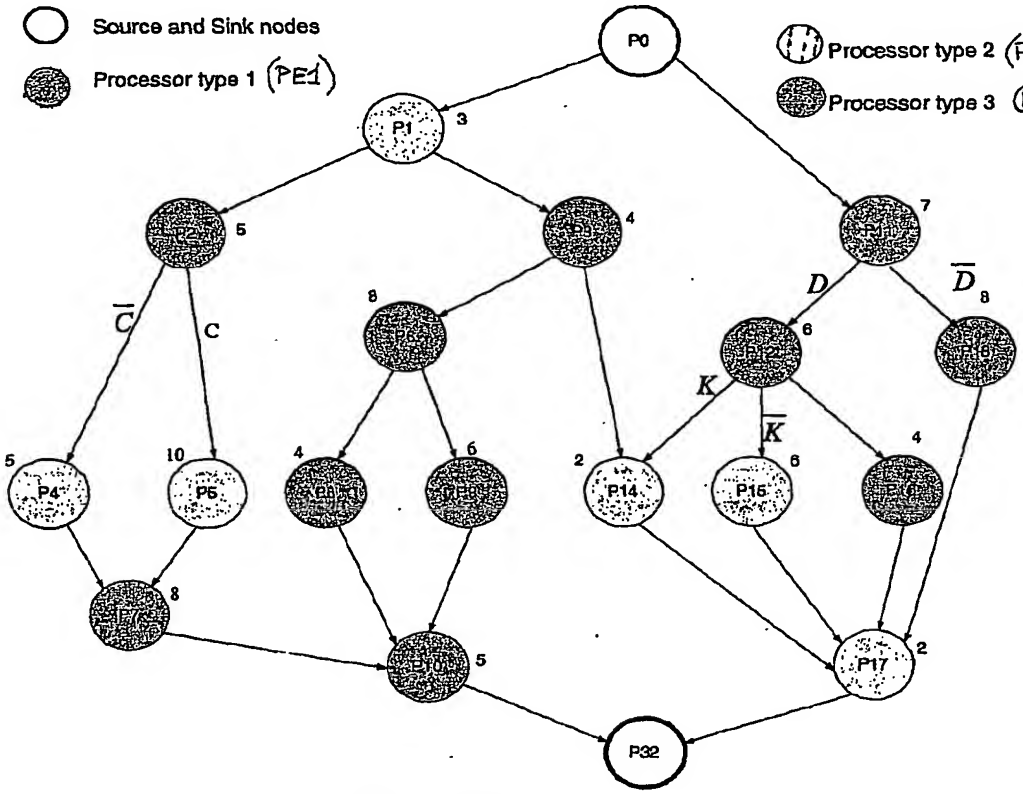


Figure 12: Path based edge activation

~~Illustrative example for our scheduling algorithm~~

A This example demonstrates the initialization strategy. It describes how the CDFG is split into individual DFGs. Moreover, it also shows the various fields required for each node and edge.

~~A. Initial CDFG:~~



For the CDFG of Fig. 13x,

B. Initialization of CDFG data structure and Branching tree proceeds as follows:

Var_indices: var[0] = D; var[1] = C; var[2] = K;
Assume number of processing elements of type = 1
Branching tree paths: DCK, DCK', DC'K, DC'K', D'CK, D'CK', D'C'K, D'C'K'
Branching tree paths not possible: D'CK, D'CK', D'C'K, D'C'K'
Removing K we get: D'C, D'C'
Final Branching tree paths: DCK, DCK', DC'K, DC'K', D'C, D'C'.

Tables xx and yy are the node and edge lists, respectively, for the CDFG of Fig. 13x. Figs. 14x - 19x are the individual Data Flow Graphs (DFGs) of the CDFG of Fig. 13x.

C. List of individual DFGs:

DFG[0] → DCK

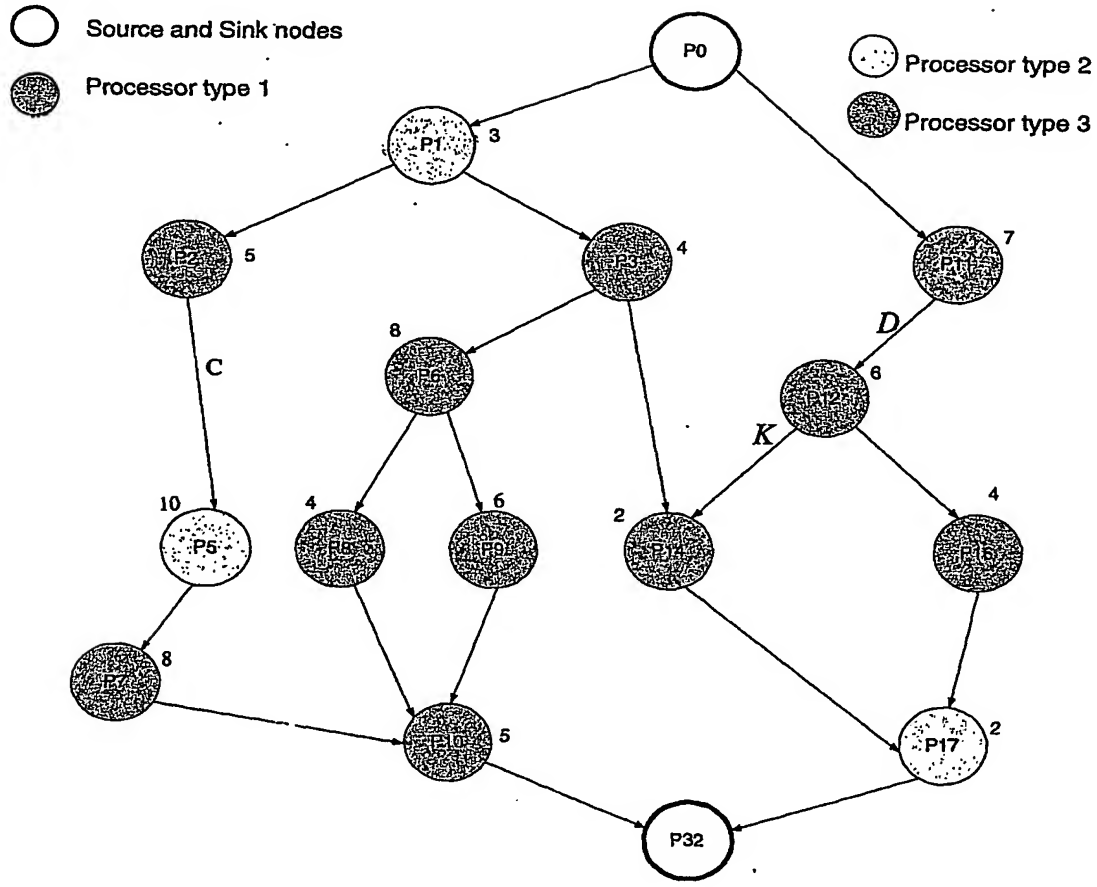


Fig 14x

DFG[1] → DCK'

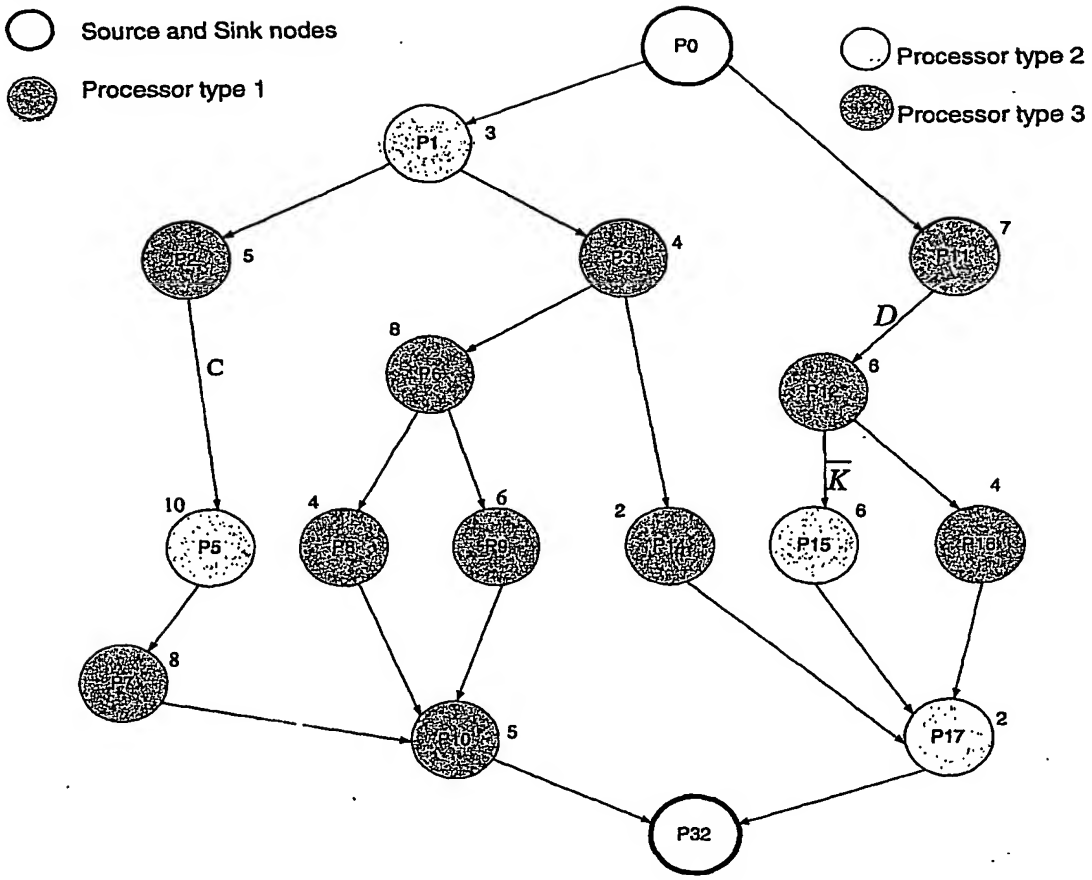


Fig. 15x

DFG[2] → DC'K

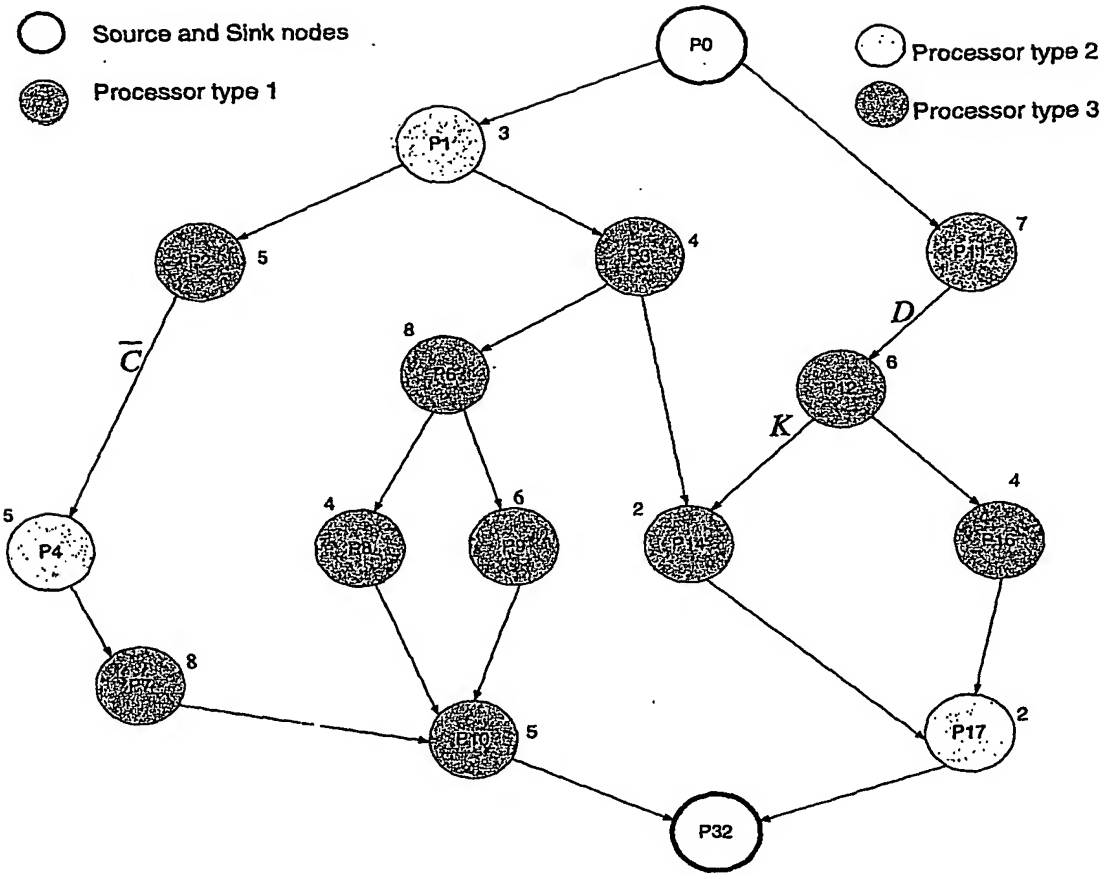


Fig. 16X

DFG[4] → D'C

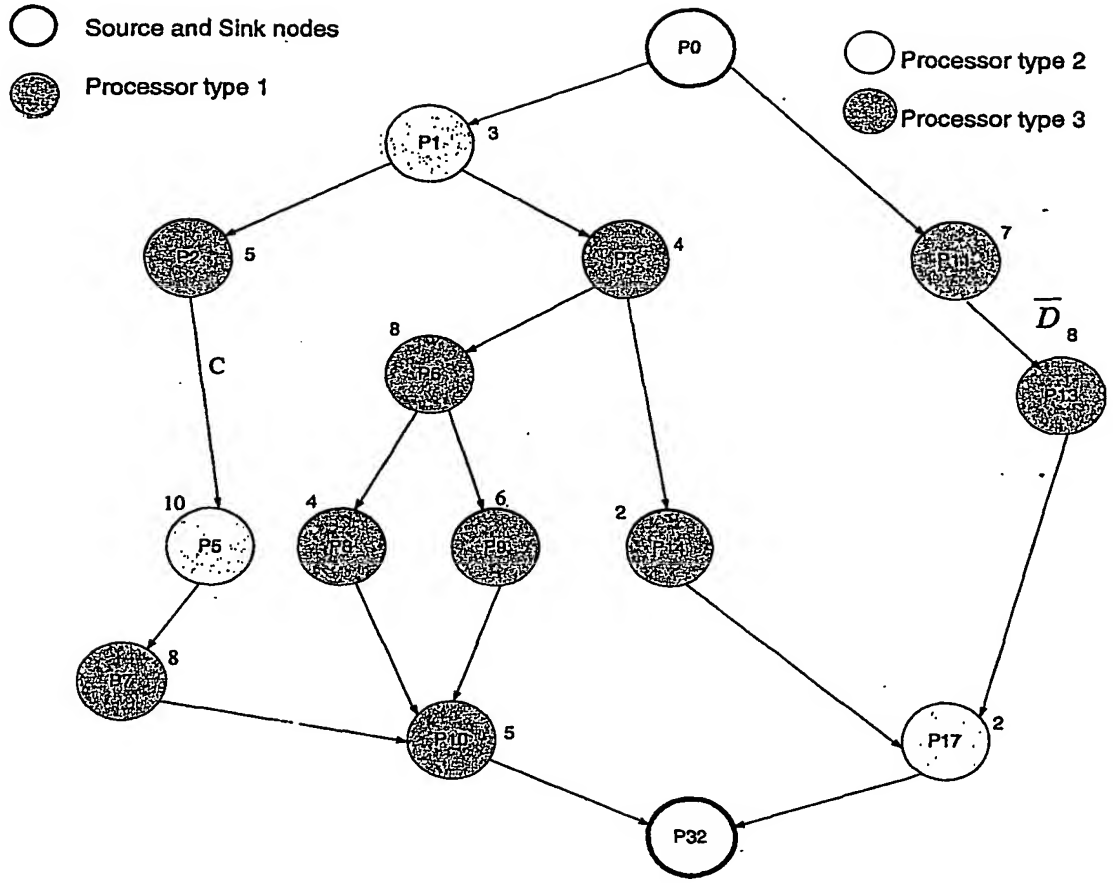


Fig 18x

DFG[5] → D'C'

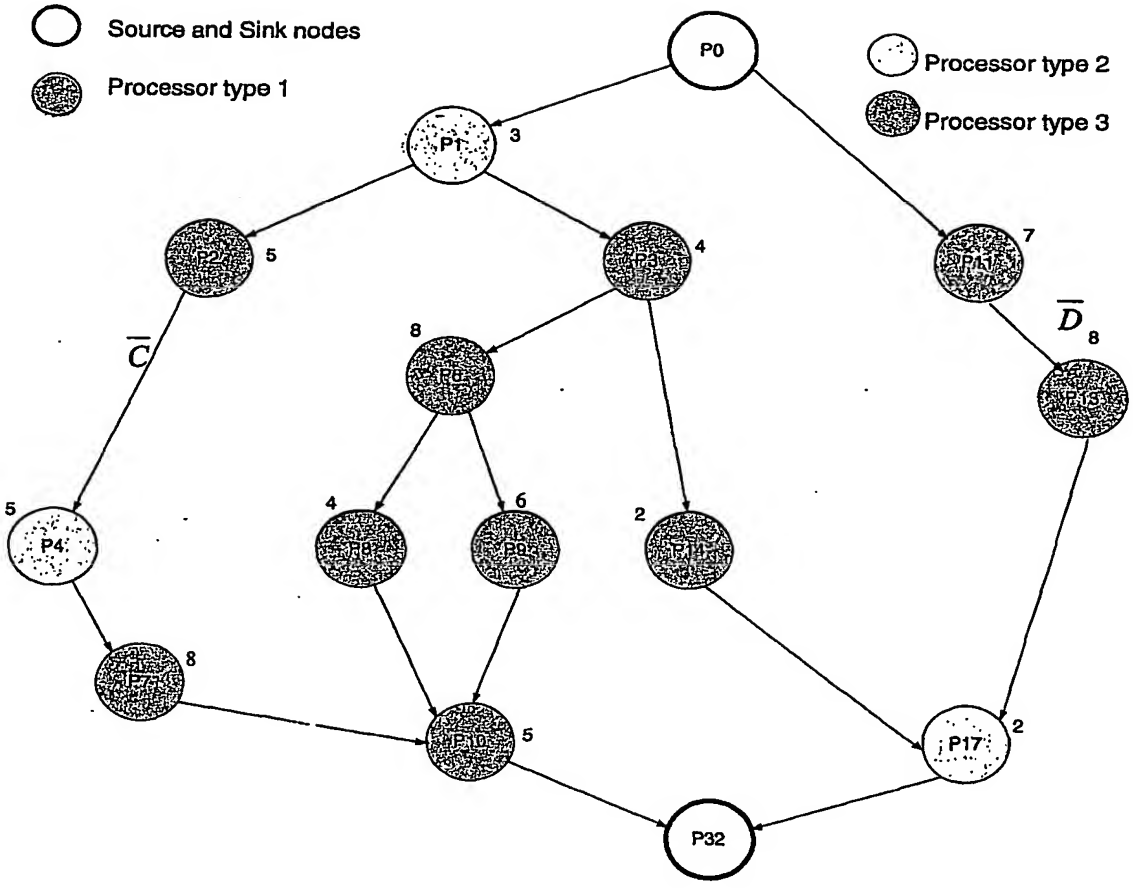


Fig 19x

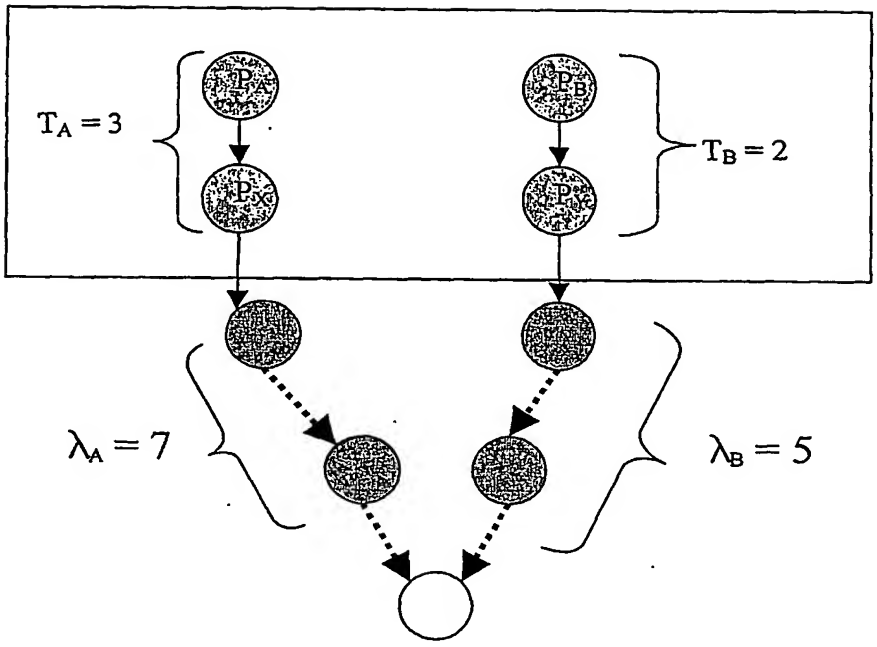
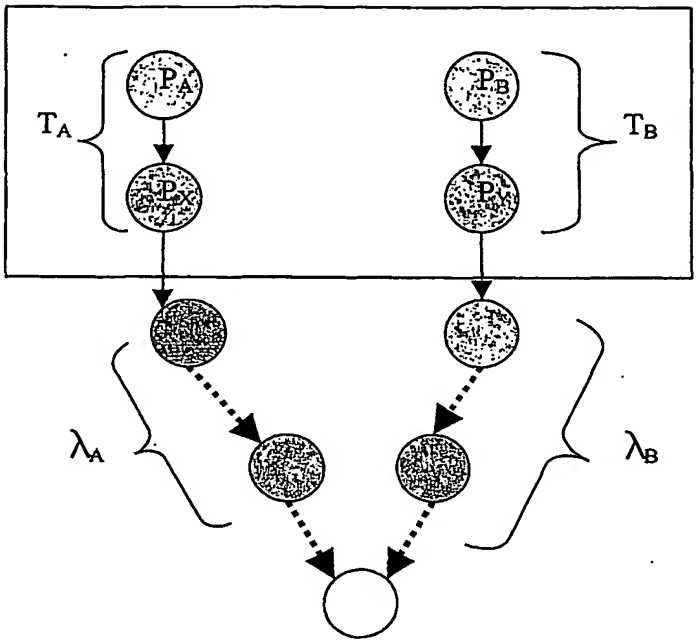
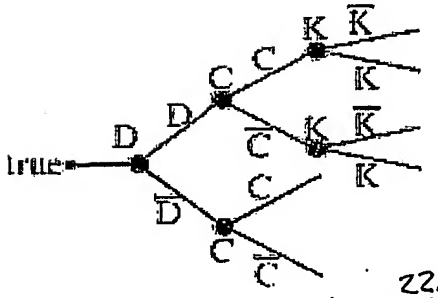


Figure ~~15~~ 20X: PCP Scheduling with Resource Dependencies in the Partial Path Region



21x

Figure 14: PCP based Scheduling



22x

Figure 16: Branching Tree

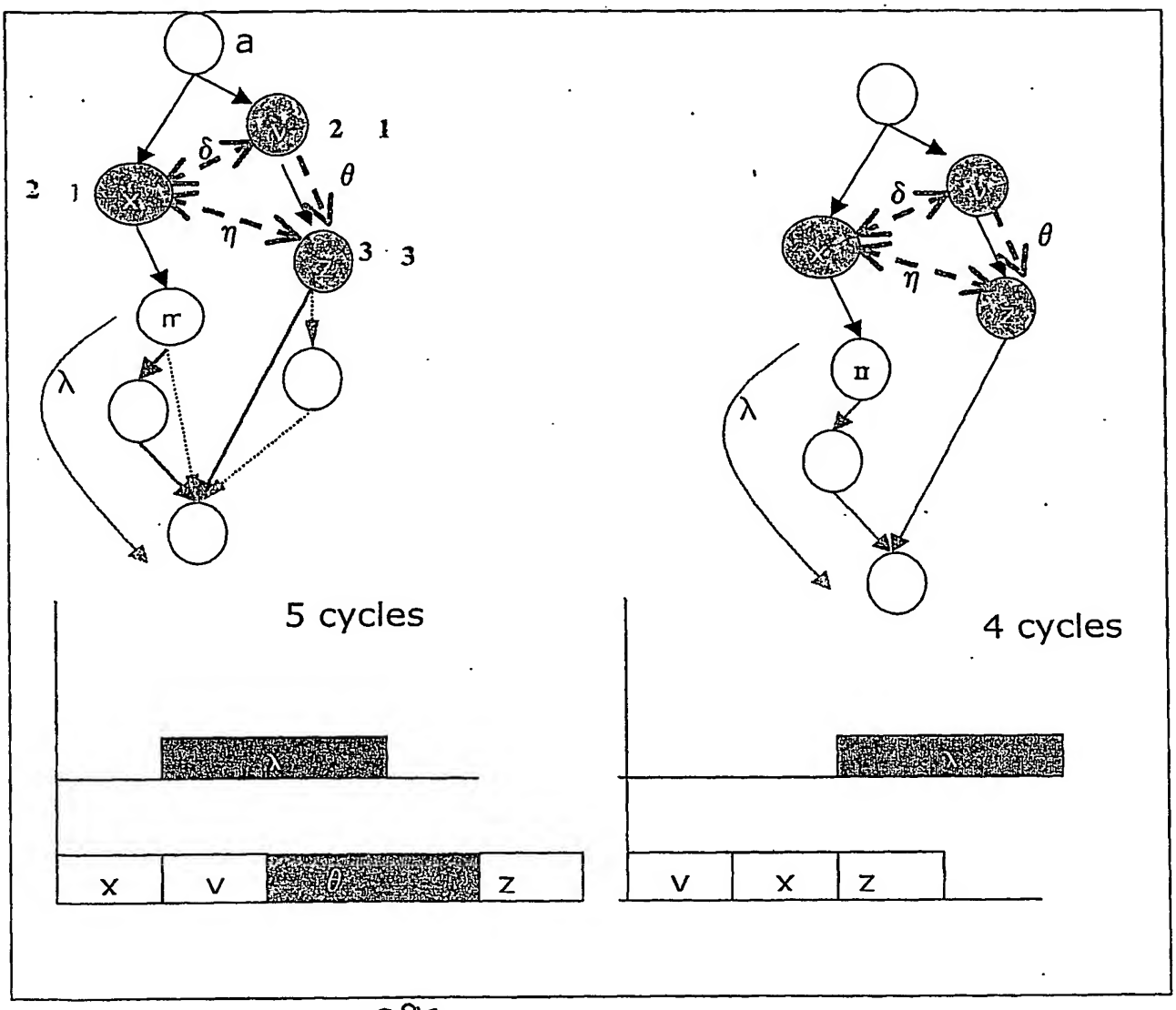


Figure 23X: Influence of Reconfiguration time on Scheduling

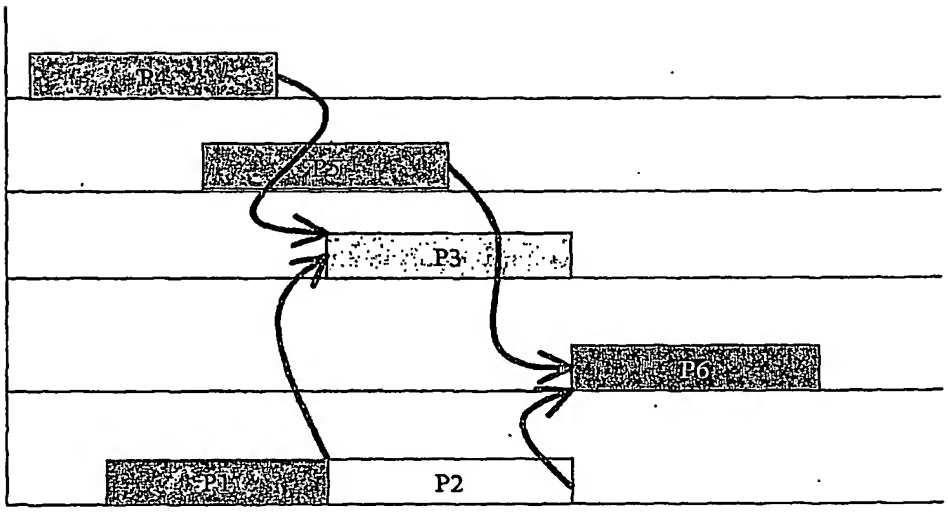


Figure ~~18~~ 24X: Scheduled Process Charts with Resource and Data Dependency

	Expression α	Expression β	Expression θ	Expression γ
Process A		0		
Process B		10		
.....				
.....				

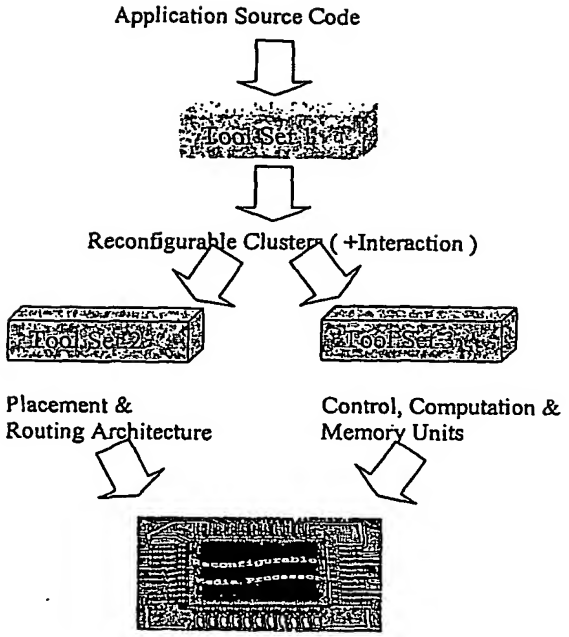
	Expression α	Expression β	Expression θ	Expression γ
Process A		30		
Process B		40		
.....				
.....				

.....

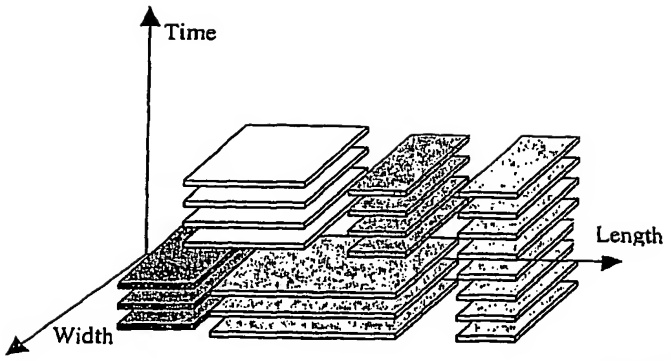
↓ and so on.

25X

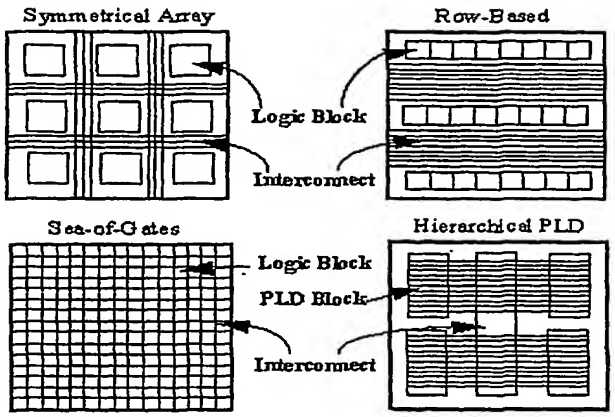
Figure ~~19~~: Dynamic Entry Updates in the NSM and LSMs



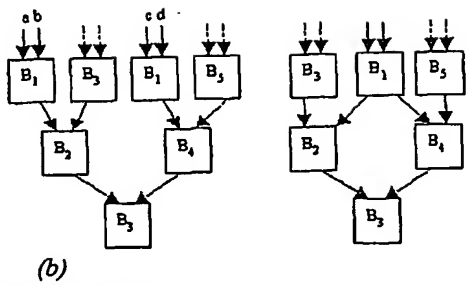
Ali-1 Tool Set Overview



Ali-2 Constraints

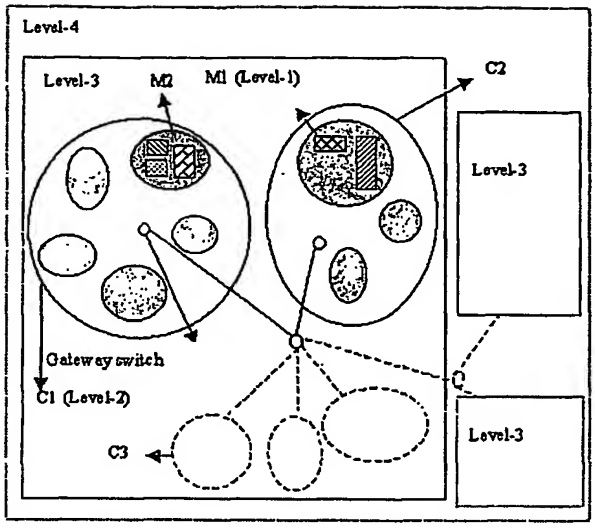


Ali-*3 Routing Architecture Overview*

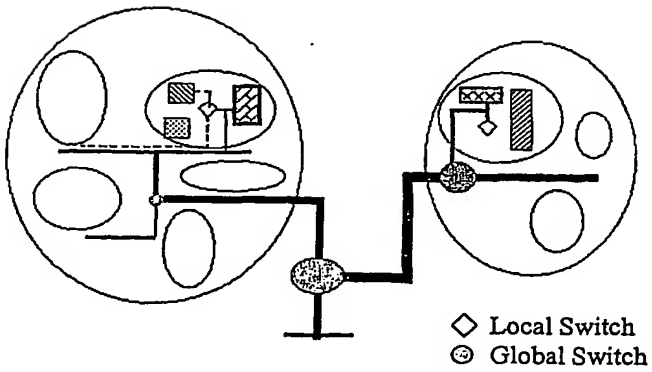


(a) *(b)*

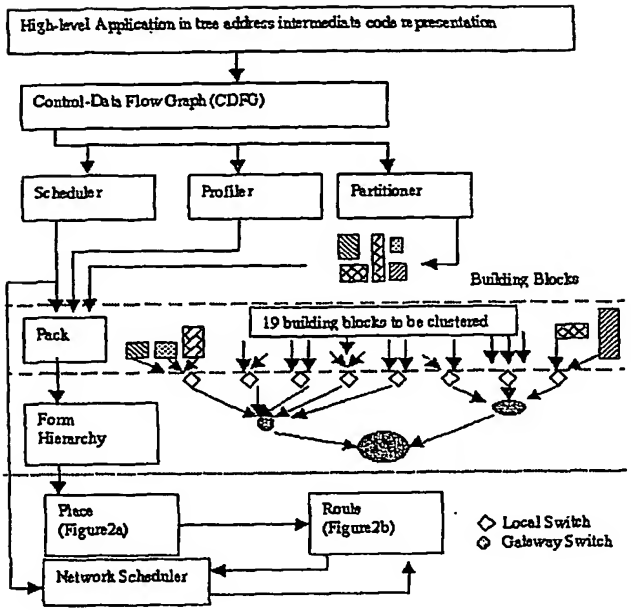
Ali-*4 Multiple vs. Single Building Block*



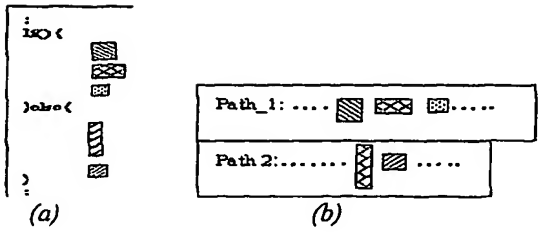
Ali-5 Overall Architecture



Ali-6 Switching



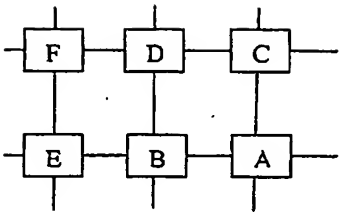
Ali-7 Methodology



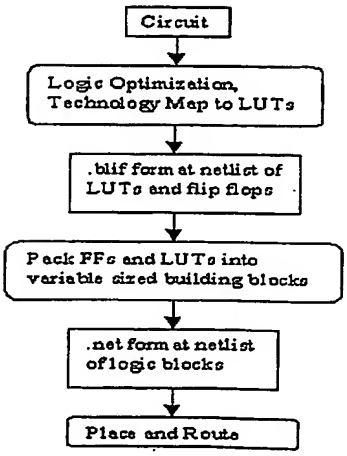
Ali-8 Control Flow Effect on Clusters

	A	B	C	D	E	F
A	0	X	X	X	X	X
B	5	0	X	X	X	X
C	6	0	0	X	X	X
D	4	3	7	0	X	X
E	1	4	1	0	0	X
F	3	0	4	5	3	0

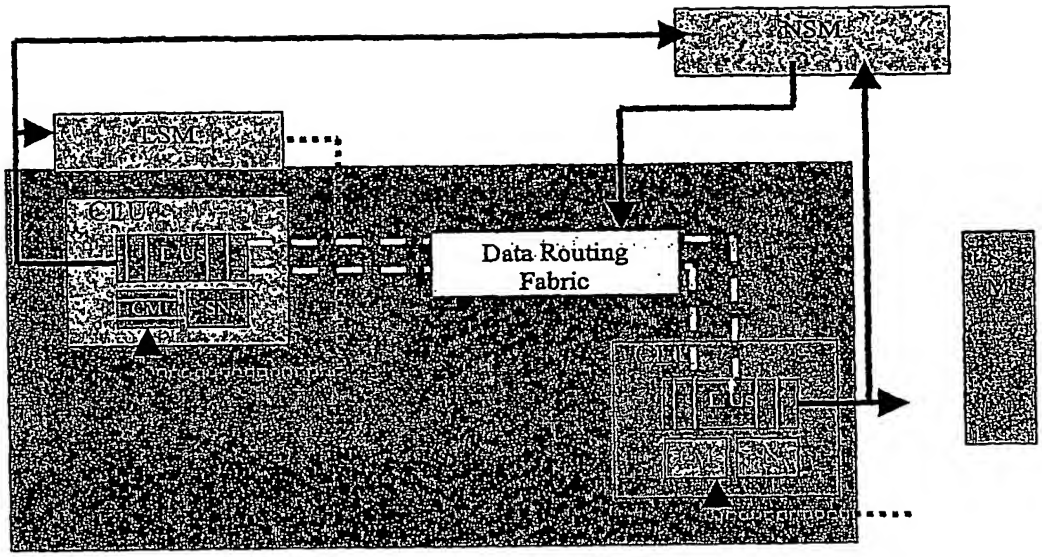
Ali-9a Cost matrix



Ali-9b Pre-placement



Ali-10 Design Flow



CLU = Configurable Logic Unit; LU = Logic Units; SN = Switching Network
CM = Configuration Memory; LSM = Logic Schedule Manager

Figure 21: The Internals of the Reconfigurable Unit

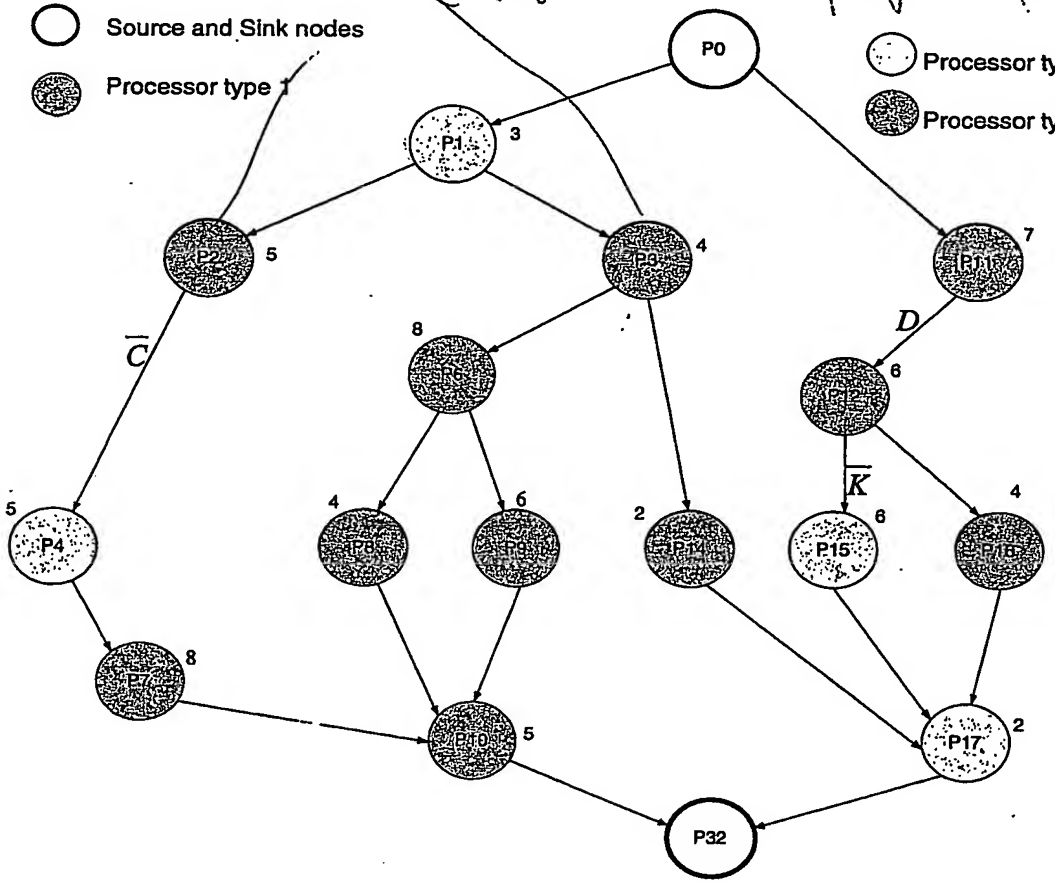

DFG[3] = DC'K'
Source and Sink nodes
Processor type 1
Processor type 2
Processor type 3
P0
P1
3
P2
5
P3
4
P11
7
$\overline{C}$
D
8
P5
6
P12
$\overline{K}$
5
P4
4
P6
6
P9
2
P14
6
P15
4
P16
8
P7
P10
5
P17
2
P32

Fig 17x